SOFTWARE TOOLS FOR TIME-DEPENDENT DIFFERENTIAL EQUATIONS

M. Berzins , P.M. Dew
(Department of Computer Studies, University of Leeds)

and

R.M. Furzeland
(Koninkliijke/Shell Laboratorium, Amsterdam)

Mathematical modelling within the oil and gas industries frequently involves the solution of complex systems of time-dependent ordinary and partial differential equations. The development of suitable mathematical models can be facilitated by the provision of a wide range of numerical methods to allow easy experimentation in the model development. The most suitable numerical methods can then be used in production runs involving the finished model. This paper is concerned with the SPRINT software of Berzins, Dew and Furzeland (1986) which has been designed to allow complex mathematical models to be integrated numerically in an efficient manner by allowing the user to choose the numerical methods to fit the characteristics of the underlying problem. The software is based on the well-known method of lines as this allows the sophisticated integrators that have been developed for initial value differential-algebraic equations to be applied to coupled systems of time dependent o.d.e.s. and p.d.e.s..

1. INTRODUCTION

The SPRINT package (Software for PRoblems IN Time), is a general-purpose computer program for the numerical solution of mathematical models that involve mixed systems of time-dependent algebraic, ordinary and partial differential equations (o.d.e.s. and p.d.e.s.). The software is the result of research between Shell Research Limited and the Department of Computer Studies at Leeds University. The aim of the research is to provide a flexible and open-ended software package to enable a user to solve a wide range of problems within a single framework. The software, as summarised by Berzins, Dew and Furzeland (1986), is based on the design philosophy that there should be a set of well-defined and independent modules that together from a "tool kit" for solving time dependent ordinary and partial differential equations.

2. SOLVING DIFFERENTIAL-ALGEBRAIC EQUATIONS USING SPRINT PROBLEM CLASS

2.1 Differential-algebraic equations problem class

The core of the software package is a versatile set of four differential-algebraic integrators, see Berzins, Dew and Furzeland (1986), with the flexibility to deal with stiff or non-stiff d.a.e.s. coupled with algebraic equations and full/banded/sparse Jacobian matrices computed analytically or numerically. Each integrator is designed to solve the class of o.d.e. initial value problems defined by

$$\underline{f}(\dot{\underline{Y}}, \underline{Y}, t) = \underline{g}(\underline{Y}, t) - A(\underline{Y}, t)\,\dot{\underline{Y}} = 0. \qquad (2.1.1)$$

with the initial condition

$$\underline{Y}(0) = \underline{K}, \qquad (2.1.2)$$

The sparse matrix A may be singular indicating a differential-algebraic system of equations. In the case when the matrix A is the identity matrix the system is said to be in normal form.

The advantage of the problem class defined by equation (2.1.1) over fully implicit o.d.e. problems is that equation (2.1.1) is linear with respect to the time derivative, i.e.

$$\partial\, \underline{f} \,/\, \partial\, \dot{\underline{Y}} = - A(\underline{Y}, t). \qquad (2.1.3)$$

All the time integration methods in SPRINT require the frequent calculation of the matrix-vector product

$$\partial\, \underline{f} \,/\, \partial\, \dot{\underline{Y}}\, \underline{z}(t)$$

where $\underline{z}(t)$ is a vector generated by the time integrator. The only exception is when the SPGEAR module, see Section 2.2, does not use the error estimator of Petzold (1982). The problem definition interface in the software requires, see equation (2.1.5) the definition of the matrix-vector product $-A(\underline{Y}, T)\,\dot{\underline{Y}}$ and so, given equation (2.1.3), can be used to economically calculate the required matrix vector product by substituting $\underline{z}(t)$ for $\dot{\underline{Y}}$ in the call to the problem definition routine. As this matrix vector product is the only substantial algorithmic difference from solving normal form problems in which A = 1 it is then possible to provide codes which are almost as efficient as those for normal form problems.

It is also required that the parts of the residual that have a $\dot{\underline{y}}$ dependence can be computed by a call to RESID and also returned to SPRINT via the vector $\underline{r}$, i.e.

$$\underline{r} = -A(\underline{y}, t)\ \dot{\underline{y}} \qquad (2.1.5)$$

One of the parameters that SPRINT passes into the RESID routine is the integer IRES; if this is set to 1 then the user must supply the form of the residual defined by equation (2.1.4), and if it is set to -1 equation (2.1.5) must be used. The integration may be interrupted by the user changing the value of IRES in RESID to force the integrator to either stop the integration, reduce the time-step to avoid a physically impossible solution value or to terminate the current step and enter the MONITR routine.

2.2 Tank level control problem

The following test problem illustrates some of the difficulties in solving differential algebraic equations. The following system of 8 algebraic-differential equations represents the control problem of keeping the level of fluid in a tank constant when a fluid flows into it. The problem arises as part of a much larger simulation of a chemical plant.

$$\dot{y}_1 = y_8 - y_4,\ y_1(0) = 50.0$$

$$\dot{y}_2 = 0.1y_6,\ y_2(0) = 0.5$$

$$\dot{y}_3 = 0.05\ (y_5 - y_3),\ y_3(0) = 0.5$$

$$0 = y_4 - 20.0\ (1 - y_3),$$

$$0 = y_5 - y_2 - y_6,$$

$$0 = y_6 - 0.5 + y_7,$$

$$0 = y_7 - 0.01\ y_1,$$

$$0 = y_8 - f(t),$$

The initial values of the algebraic variables y_3, y_4, y_5, y_6 and y_7 are consistent with the initial values of the

past the discontinuities in the first derivative of the input flow on runs 3, 4 and 5. The STHETB code also competes at low to medium accuracy requirements with the more widely used SPGEAR backward differentiation code.

2.3 Solution of more general D.A.E.s using SPRINT

The difficulty of solving general d.a.e. systems depends critically on the index of the system. The index is defined by Petzold and Lostedt (1983) as the number of times the algebraic constraints must be differentiated in order to obtain an o.d.e. system in normal form. Thus the simplest index i problem is defined by

$$y_1(t) = g(t),\ y_j = \dot{y}_{j-1},\ j = 2,\ldots,i \qquad (2.3.1)$$

which has the solution

$$y_1(t) = g(t),\ y_j = \frac{d^{j-1}g}{dt^{j-1}},\ j = 2,\ldots,i. \qquad (2.3.2)$$

The index is a measure of the singularity of the system. A standard o.d.e. problem in normal form has index 0 while the tank level control problem above has index 1.

Petzold and Gear (1984) have investigated applying backward differentiation formulae such as those used in SPRINT, to d.a.e.s. and have shown that they can only be reliably applied to systems of index 2 or less or to index 3 problems of a special form.

The only applied technique for problems of index 3 and above is to use the backward Euler method and extrapolation, as in the POST code of Schryer (1984), but there are still formidable practical difficulties to be overcome, particularly in solving the ill-conditioned non-linear equations that may arise, see Petzold and Gear (1984). Problems of arbitrarily high index can arise for instance, in electronic circuit modelling which is one of the applications POST was designed for.

3. SOLVING PARTIAL DIFFERENTIAL EQUATIONS USING SPRINT

In the Method of Lines the partial differential equations are spatially discretised over NPTS points using finite-difference, finite element or collocation methods. This discretisation results in a system of NPTS non-linear, coupled d.a.e.s. for each given p.d.e. and provides a unified approach to solving mixed systems of d.a.e.s. and p.d.e.s..

3.1 The SPRINT spatial discretisation routines

There are currently two discretisation modules in SPRINT: SPDIFF a lumped finite element method developed by Skeel and Berzins (1987) and SGENCO, the C^0 collocation discretisation discussed by Berzins, Dew and Furzeland (1986). Each of these modules has a setup routine which performs the initialisation tasks and which must be called before the main part of SPRINT is entered. The three other main components are the RESID routine discussed above, the MONITR routine that is called by SPRINT at the end of every time-step and an interpolation routine that can be used to generate extra solution values after SPRINT has integrated to the required time.

In this paper we shall only use the SPDIFF discretisation method which is analogous to the usual central, three-point finite-difference formula for problems in Cartesian co-ordinates. However for problems in polar and spherical co-ordinates the three-point formula is suitably modified to maintain second-order accuracy. An option is provided within this module to allow the user to adaptively vary the space mesh in time, see Berzins, Dew and Furzeland (1986).

3.2 Pool evaporation problem

This section provides an example of a non-standard problem that has been solved very effectively using the SPDIFF discretisation and the SPGEAR integrator from the SPRINT software. The problem concerns the rate of evaporation of vapour from a pool of liquid of length one metre. Above the pool a constant (i.e. non-time varying) wind blows. There is a viscous sub-layer above the pool of height $\bar{x}$ and above this is a 'windy' region in which the concentration of vapour diminishes until it is negligible at a height of about $10^3\ \bar{x}$. In order to apply the method of lines to this problem we take the spatial variable as being the height above the pool and integrate across the length of the pool.

The governing p.d.e. for the vapour concentration $C(x, t)$ in the viscous sub-layer is

$$6.8\times10^{3}\ x\ \frac{\partial C}{\partial t} = \frac{\partial}{\partial x}\left(8.65\times10^{-6}\ \frac{\partial C}{\partial x}\right),\ x \in [0,\ \bar{x}] \qquad (3.2.1)$$

and in the turbulent region above the viscous sub-layer

$$(0.7717\ \log(x) + 9.313)\ \frac{\partial C}{\partial t} = \frac{\partial}{\partial x}\left(0.1297\ x\ \frac{\partial C}{\partial x}\right), x \in [\bar{x},\ 1.0]\ . \qquad (3.2.2)$$

The problem has the following non-standard feature. As a check on the accuracy of the numerical solution it is important to compare the rate of evaporation Q_1 at the surface of the pool with the quantity of vapour which passes above a given point in the pool, Q_2 where Q_1 and Q_2 are defined by

$$Q_1(t) = -7.934\text{x}10^{-7} \int_0^t \frac{\partial C}{\partial x}(0, t)\, dt \qquad (3.2.5)$$

and

$$Q_2(t) = 9.4175\text{x}10^{-2} \int_0^1 u(x)\, C(x, t)\, dx \qquad (3.2.6)$$

where the function $u(x)$ is defined by

$$u(x) = 6.81\text{x}10^{3}\, x,\ x \in [0, \bar{x}] \qquad (3.2.7)$$

$$= (0.7717 \log(x) + 9.313),\ x \in [\bar{x}, 1.0] \qquad (3.2.8)$$

This comparison is achieved by defining an extra coupled o.d.e. for the rate of evaporation

$$\frac{dQ_1}{dt} = -7.934\text{x}10^{-7} \frac{\partial C}{\partial x}(0, t). \qquad (3.2.9)$$

Similarly we approximate equation (3.2.6) by using trapezoidal quadrature in space to get

$$Q_2(t) = 9.4175\text{x}10^{-2} \sum_{i=1}^{NPTS} u(x_i)\, C(x_i, t)\, W_i \qquad (3.2.10)$$

where W_i are the trapezoidal quadrature weights for the unequally spaced data and NPTS is the number of spatial mesh points. The vapour discrepancy in the calculation can then be monitored continuously by defining a new variable $Q_3(t)$ by

$$Q_3(t) = Q_2(t) - Q_1(t) \qquad (3.2.11)$$

Equations (3.2.9), (3.2.10) and (3.2.11) are then integrated with the p.d.e. as a mixed p.d.e./o.d.e. system. The initial conditions for the new variables are $Q_i(0) = 0$ for $i = 1,2,3$. An inspection of the values of $Q_3(t)$

4. CONCLUSIONS

Our experience of using SPRINT in an industrial environment has led to the following observations on its advantages

(1) Ability to handle a general problem formulation

  (a) mixed o.d.e.s. plus algebraic equations, as in Section 2.2 above

  (b) mixed o.d.e.s. plus p.d.e.s., as in Section 3 above

  (c) general non-linear form of governing equations, e.g. the bubble collapse problem in Berzins, Dew and Furzeland (1986).

(2) Fast software development times - coding times for development models are greatly reduced and allow the scientist to concentrate more on the physics of the problem.

(3) Flexibility in software building - the software can be readily adapted to other solution methods by interchanging the time integrator, the spatial discretisation method or the MONITR routine. This is especially helpful in research and development work.

(4) Allows the user easy access to advanced numerical techniques

  (a) stiff, non-stiff and type insensitive time integrators

  (b) efficient time stepping, especially for d.a.e.s.

  (c) capability to handle discontinuities in the time integration

  (d) full, banded or sparse linear algebra

  (e) adaptive space remeshing with finite difference formulae

  (f) high order collocation approximation in space.

The software needs to be extended to automatically discretise problems in two and three space dimensions. This raises the question of how the very large systems of non-

Berzins, M. and Furzeland, R.M., (1986), "A Type-Insensitive Method for the Solution of Stiff and Non-Stiff Differential Equations", Leeds University, Department of Computer Studies, Report No: 204.

Brown, P.N. and Hindmarsh, A.C., (1986), "Matrix-Free Methods for Stiff Systems of O.D.E.s, SIAM, *J. Numer. Anal.*, Vol. 23, Np: 3, pp 610-638.

Chua, T.S. and Dew, P.M., (1984), "The Design of a Variable-Step Integrator for the Simulation of Gas Transmission Networks, *Int. Jour. for Num. Meths. in Engineering*, Vol. 20, pp 1797-1813.

Hindmarsh, A.C., (1982), "ODEPACK - A Systematized Collection of O.D.E. Solvers", Lawrence Livermore Nat. Lab., Report No: 88007, also p.312 of Advances in Computer Methods IV, Vichnevetsky, R. and Stepleman, R.S. (eds.), IMACS (int. Assoc. Maths. Computers in Simulation), New Brunswick.

Petzold, L., (1982), "Differential Algebraic Equations are not O.D.E.s.", *S.I.A.M. J. Sci. Stat. Comput.*, Vol. 3, pp 367-384.

Petzold, L. and Gear, C.W., (1984), "O.D.E. Methods for the Solution of Differential - Algebraic Systems", *S.I.A.M. Journal Numerical Analysis*, Vol. 21, No: 4, August.

Petzold, L. and Lostedt, P., (1986), "Numerical Solution of Non-linear Differential Equations with Algebraic Constraints II - Practical Implications", *S.I.A.M. Journal of Sci. and Stat. Computing.*

Prothero, A. and Robinson, A., (1974), On the Accuracy and Stability of One Step Methods for Solving Stiff Systems of O.D.E.s., *Math. Comp.*, 28, 145.

Schryer, N., (1984), "Partial Differential Equations in One Space Variable", Computing Science Technical Report No: 115, AT and T Laboratories, Murray Hill, New Jersey 07974.

Skeel, R.D. and Berzins, M., (1987), "Spatial Discretisation Methods for Parabolic P.D.E.s., Report 217, Department of Computer Studies, Leeds University, Leeds LS2 9JT.